

4/30/24, 4:04 PM

Buttons | Apple Developer Documentation

Design

Overview

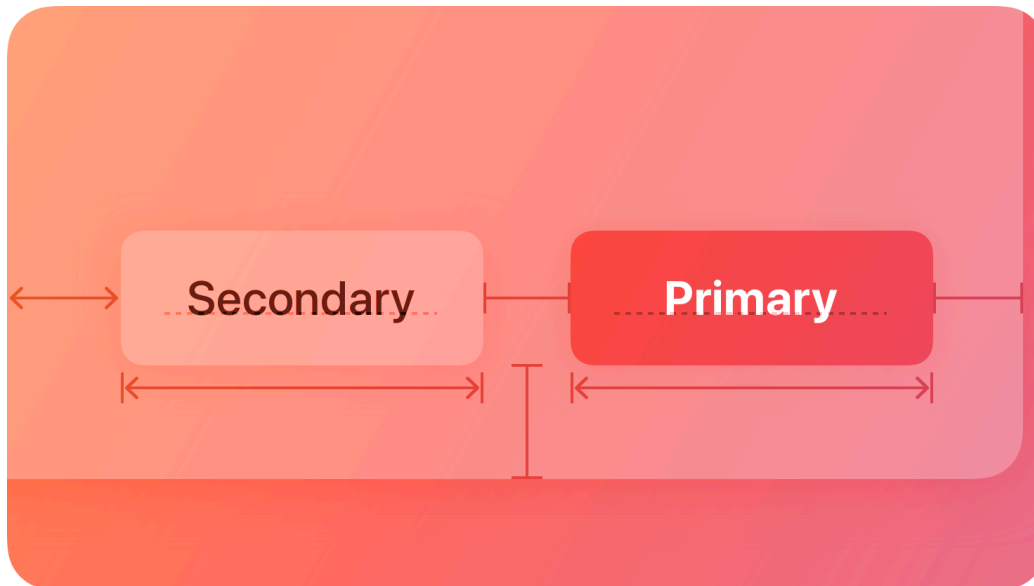
What's new

Guidelines

Resources

Buttons

A button initiates an instantaneous action.



Versatile and highly customizable, buttons give people simple, familiar ways to do tasks in your app. In general, a button combines three attributes to clearly communicate its function:

- **Style.** A visual style based on size, color, and shape.
- **Content.** The symbol (or interface icon), text label, or both that a button displays to convey its purpose.
- **Role.** A system-defined role that identifies a button's semantic meaning and can affect its appearance.

In addition to all-purpose buttons, many common button styles — like Info, Close, and Contact Add — perform familiar tasks throughout the system. There are also many button-like components that have distinct appearances and behaviors for specific use cases, like toggles, pop-up buttons, and segmented controls.

Best practices

When buttons are instantly recognizable and easy to understand, an app tends to feel intuitive and well designed.

Make buttons easy for people to use. It's essential to include enough space around a button so that people can visually distinguish it from surrounding components and content. Giving a button enough space is also critical for helping people select or activate it, regardless of the

4/30/24, 4:04 PM

Buttons | Apple Developer Documentation

method of input they use. As a general rule, a button needs a hit region of at least 44x44 pt — in visionOS, 60x60 pt — to ensure that people can select it easily, whether they use a fingertip, a pointer, their eyes, or a remote.

Ensure that each button clearly communicates its purpose. A button always includes a text label or a symbol (or interface icon) — and sometimes a combination of both — to help people predict what it does.

Style

System buttons offer a range of styles that support extensive customization while providing built-in interaction states, accessibility support, and appearance adaptation. Different platforms define different styles that help you communicate hierarchies of actions in your app.

You can configure a system button to use any combination of style and size. By default, a system button uses a size-specific corner radius and your app's accent color. If necessary, you can change these attributes, in addition to attributes that control content layout and the presence of an activity indicator.

In general, use a button that has a visible background for the most likely action in a view.

Buttons that have a fill or background shape tend to be the most visually prominent, helping people quickly identify the action they're most likely to want.

Consider keeping the number of visually prominent buttons to one or two per view. Giving people too many prominent buttons increases their cognitive load, requiring them to spend more time considering options before making a choice.

Use style — not size — to visually distinguish the preferred choice among multiple options.

When you use buttons of the same size to offer two or more options, you signal that the options form a coherent set of choices. If you want to highlight the preferred or most likely option in a set, use a more prominent button style for that option and a less prominent style for the remaining ones.

Content

Create button content that helps people instantly understand what the button does.

Depending on the platform, a button can contain an icon, text, or both.

Note

In macOS and visionOS, the system displays a tooltip after people hover over a button for a moment. A tooltip displays a brief phrase that explains what a button does; for guidance, see [Offering help](#).

Consider using an icon when a button performs a familiar action that people associate with the icon. For example, people can predict that a button containing the `square` and `arrow` up symbol will help them perform share-related activities. If it makes sense to use an icon in your button, consider using an existing or customized [symbol](#).

Consider using text when a short label communicates more clearly than an icon. To use text, write a few words that succinctly describe what the button does. Using [title-style capitalization](#),

4/30/24, 4:04 PM

Buttons | Apple Developer Documentation

consider starting the label with a verb to help convey the button's action — for example, a button that lets people add items to their shopping cart might use the label "Add to Cart."

Role

A system button can have one of the following roles:

- **Normal.** No specific meaning.
- **Primary.** The button is the default button — the button people are most likely to choose.
- **Cancel.** The button cancels the current action.
- **Destructive.** The button performs an action that can result in data destruction.

A button's role can have additional effects on the appearance you configure. For example, the system can use bold text for the label in a primary button, whereas a destructive button uses a red color.

Assign the primary role to the button people are most likely to choose. When a primary button responds to the Return key, it makes it easy for people to quickly confirm their choice. In addition, when the button is in a temporary view — like a sheet, an editable view, or an alert — assigning it the primary role means that the view can automatically close when people press Return.

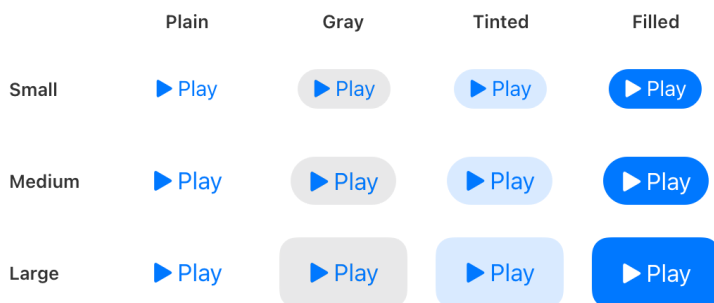
Don't assign the primary role to a button that performs a destructive action, even if that action is the most likely choice. Because of its visual prominence, people sometimes choose a primary button without reading it first. Help people avoid losing content by assigning the primary role to nondestructive buttons.

Platform considerations

No additional considerations for tvOS.

iOS, iPadOS

iOS and iPadOS have four button styles, each available in three sizes. Each combination of size and style has a different level of visual prominence.



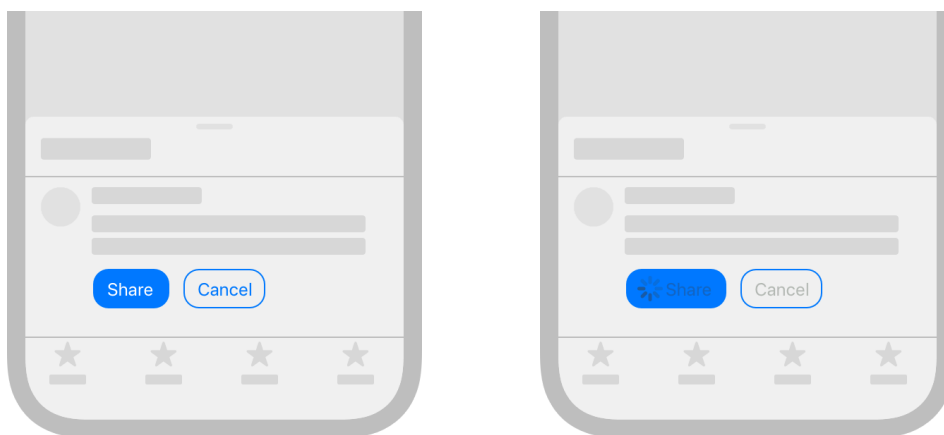
Include additional text below the label only if it provides useful details. Additional text uses a smaller text size than the label, showing that the information is secondary to the button's action. For example, you might add text to update an Add to Cart button with the number of items in the

4/30/24, 4:04 PM

Buttons | Apple Developer Documentation

cart. Avoid writing a subtitle that explains more about what the button does; make sure a button's containing view, label or image, style, and role provide all the information people need to understand its action.

Configure a button to display an activity indicator when you need to provide feedback about an action that doesn't instantly complete. Displaying an activity indicator within a button can save space in your user interface while clearly communicating the reason for the delay. To help clarify what's happening, you can also configure the button to display a different label alongside the activity indicator. For example, the label "Checkout" could change to "Checking out..." while the activity indicator is visible. When a delay occurs after people click or tap your configured button, the system displays the activity indicator next to the original or alternative label, hiding the button image, if there is one.



macOS

Several specific button types are unique to macOS.

Push buttons

The standard button type in macOS is known as a *push button*. You can configure a push button to display text, a symbol, an interface icon, or an image, or a combination of text and image content. Push buttons can act as the default button in a view and you can tint them.

Use a flexible-height push button only when you need to display tall or variable height content. Flexible-height buttons support the same configurations as regular push buttons — and they use the same corner radius and content padding — so they look consistent with other buttons in your interface. If you need to present a button that contains two lines of text or a tall icon, use a flexible-height button; otherwise, use a standard push button. For developer guidance, see [`NSButton.BezelStyle.flexiblePush`](#).

Append a trailing ellipsis to the title when a push button opens another window, view, or app. Throughout the system, an ellipsis in a control title signals that people can provide additional input. For example, the Edit buttons in the AutoFill pane of Safari Settings include ellipses because they open other views that let people modify autofill values.

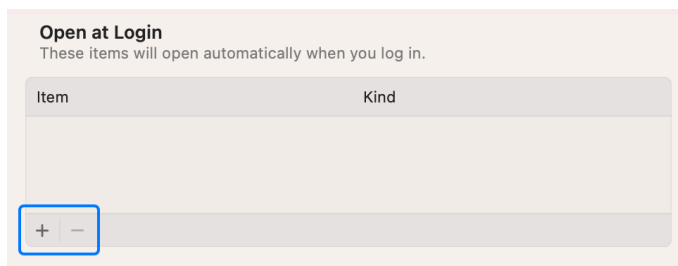
Consider supporting spring loading. On systems with a Magic Trackpad, *spring loading* lets people activate a button by dragging selected items over it and force clicking — that is, pressing harder — without dropping the selected items. After force clicking, people can continue dragging the items, possibly to perform additional actions.

4/30/24, 4:04 PM

Buttons | Apple Developer Documentation

Gradient buttons

A gradient button initiates an action related to a view, like adding or removing rows in a table.



Gradient buttons contain symbols or interface icons — not text — and you can configure them to behave like push buttons, toggles, or pop-up buttons. The buttons appear in close proximity to their associated view — usually within or beneath it — so people know which view the buttons affect.

Use gradient buttons in a view, not in the window frame. Gradient buttons aren't intended for use in toolbars or status bars. If you need a button in a toolbar, use a toolbar item.

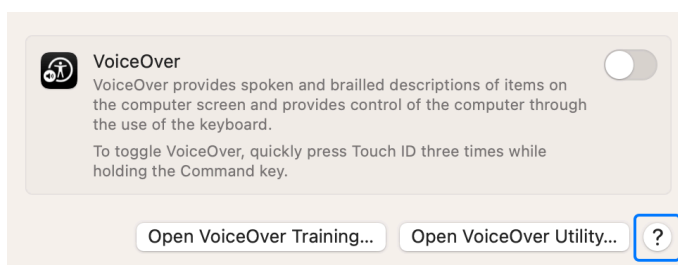
Prefer using a symbol in a gradient button. SF Symbols provides a wide range of symbols that automatically receive appropriate coloring in their default state and in response to user interaction.

Avoid using labels to introduce gradient buttons. Because gradient buttons are closely connected with a specific view, their purpose is generally clear without the need for descriptive text.

For developer guidance, see `NSButton.BezelStyle.smallSquare`.

Help buttons

A help button appears within a view and opens app-specific help documentation.



Help buttons are circular, consistently sized buttons that contain a question mark. For guidance on creating help documentation, see Offering help.

Use the system-provided help button to display your help documentation. People are familiar with the appearance of the standard help button and know that choosing it opens help content.

When possible, open the help topic that's related to the current context. For example, the help button in the Rules pane of Mail settings opens the Mail User Guide to a help topic that explains how to change these settings. If no specific help topic applies directly to the current context, open the top level of your app's help documentation when people choose a help button.

4/30/24, 4:04 PM

Buttons | Apple Developer Documentation

Include no more than one help button per window. Multiple help buttons in the same context make it hard for people to predict the result of clicking one.

Position help buttons where people expect to find them. Use the following locations for guidance.

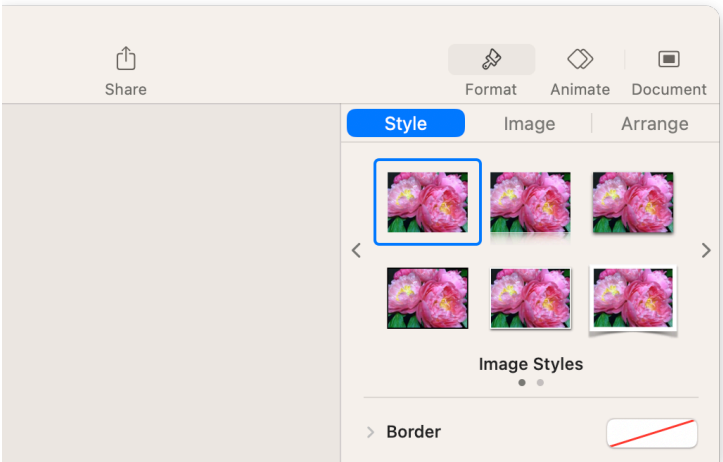
View style	Help button location
Dialog with dismissal buttons (like OK and Cancel)	Lower corner, opposite to the dismissal buttons and vertically aligned with them
Dialog without dismissal buttons	Lower-left or lower-right corner
Settings window or pane	Lower-left or lower-right corner

Use a help button within a view, not in the window frame. For example, avoid placing a help button in a toolbar or status bar.

Avoid displaying text that introduces a help button. People know what a help button does, so they don't need additional descriptive text.

Image buttons

An image button appears in a view and displays an image, symbol, or interface icon. You can configure an image button to behave like a push button, toggle, or pop-up button.



Use an image button in a view, not in the window frame. For example, avoid placing an image button in a toolbar or status bar. If you need to use an image as a button in a toolbar, use a toolbar item. See [Toolbars](#).

Include about 10 pixels of padding between the edges of the image and the button edges. An image button's edges define its clickable area even when they aren't visible. Including padding ensures that a click registers correctly even if it's not precisely within the image. In general, avoid including a system-provided border in an image button; for developer guidance, see [is Bordered](#).

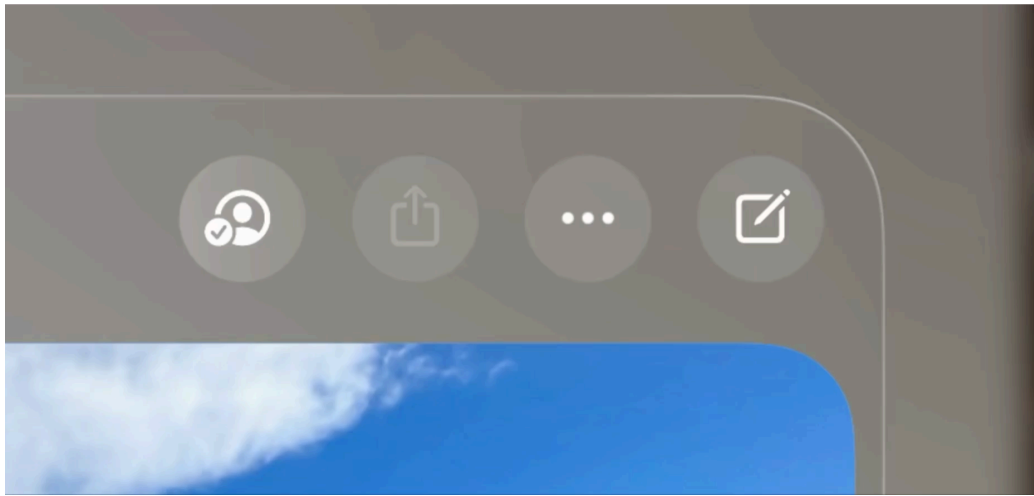
If you need to include a label, position it below the image button. For related guidance, see [Labels](#).

4/30/24, 4:04 PM

Buttons | Apple Developer Documentation

visionOS

A visionOS button typically includes a visible background that can help people see it, and the button plays sound to provide feedback when people interact with it.



Play 

There are three standard button shapes in visionOS. Typically, an icon-only button uses a circle shape, a text-only button uses a roundedRectangle or capsule shape, and a button that includes both an icon and text uses the capsule shape.

visionOS buttons use different visual styles to communicate five different interaction states.



Idle



Hover



Selected



Toggled



Unavailable

Note

In visionOS, buttons don't support custom hover effects.

In addition to the five states shown above, a button can also reveal a tooltip when people look at it for a brief time. In general, buttons that contain text don't need to display a tooltip because the button's descriptive label communicates what it does.



Play ▶

In visionOS, buttons can have the following sizes.

Shape	Mini (28 pt)	Small (32 pt)	Regular (44 pt)	Large (52 pt)	Extra Large (64 pt)
Circular	✓	✓	✓	✓	✓
Capsule (text only)		✓	✓	✓	
Capsule (text and icon)			✓	✓	
Rounded rectangle		✓	✓	✓	

Prefer buttons that have a discernible background shape and fill. It tends to be easier for people to see a button when it’s enclosed in a shape that uses a contrasting background fill. The exception is a button in a toolbar, context menu, alert, or ornament where the shape and material of the larger component make the button comfortably visible. The following guidelines can help you ensure that a button looks good in different contexts:

- When a button appears on top of a glass window, use the thin material as the button’s background.
- When a button appears floating in space, use the glass material for its background.

Avoid creating a custom button that uses a white background fill and black text or icons. The system reserves this visual style to convey the toggled state.

4/30/24, 4:04 PM

Buttons | Apple Developer Documentation

In general, prefer circular or capsule-shape buttons. People's eyes tend to be drawn toward the corners in a shape, making it difficult to keep looking at the shape's center. The more rounded a button's shape, the easier it is for people to look steadily at it. When you need to display a button by itself, prefer a capsule-shape button.

Provide enough space around a button to make it easy for people to look at it. Aim to place buttons so their centers are always at least 60 points apart. If your buttons measure 60 points or larger, add four points of padding around them to keep the hover effect from overlapping. Also, it's usually best to avoid displaying small or mini buttons in a vertical stack or horizontal row.

Choose the right shape if you need to display text-labeled buttons in a stack or row. Specifically, prefer the rounded-rectangle shape in a vertical stack of buttons and prefer the capsule shape in a horizontal row of buttons.

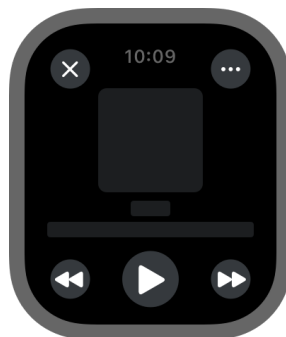
Use standard controls to take advantage of the audible feedback sounds people already know. Audible feedback is especially important in visionOS, because the system doesn't play haptics.

watchOS

watchOS displays all buttons using the capsule button shape. In watchOS 10, the buttons are larger and have material backgrounds that adapt to underlying content to ensure legibility.



Use the toolbar to place buttons in the corners. The system automatically moves the time and title to accommodate toolbar buttons.



Toolbar buttons automatically inherit the app's tint color. Only change the tint color when you need to convey important information, such as when a button performs a destructive action.

Prefer buttons that span the width of the screen for primary actions in your app. Full-width buttons look better and are easier for people to tap. If two buttons must share the same

4/30/24, 4:04 PM

Buttons | Apple Developer Documentation

horizontal space, use the same height for both, and use images or short text titles for each button's content.

Use toolbar buttons to provide either navigation to related areas or contextual actions for the view's content. These buttons provide access to additional information or secondary actions for the view's content.

Use the same height for vertical stacks of one- and two-line text buttons. As much as possible, use identical button heights for visual consistency.

Resources

Related

[Pop-up buttons](#)

[Pull-down buttons](#)

[Toggles](#)

[Segmented controls](#)

[Location button](#)

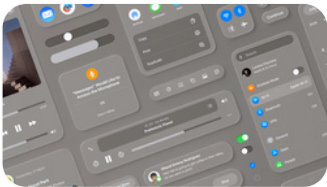
Developer documentation

[Button](#) — SwiftUI

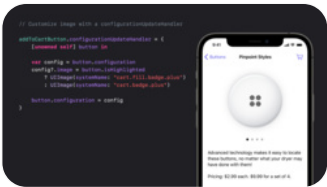
[UIButton](#) — UIKit

[NSButton](#) — AppKit

Videos



Design for spatial user interfaces



Meet the UIKit button system



What's new in UIKit

Supported platforms



- Buttons
- Best practices
- Style
- Content
- Role
- Platform considerations
- Resources
- Change log

Change log

Date	Changes
February 2, 2024	Noted that visionOS buttons don't support custom hover effects.
December 5, 2023	Clarified some terminology and guidance for buttons in visionOS.

4/30/24, 4:04 PM

Buttons | Apple Developer Documentation

Date	Changes
June 21, 2023	Updated to include guidance for visionOS.
June 5, 2023	Updated guidance for using buttons in watchOS.